

# Deep Neural Architectures for Prediction in Healthcare

## Abstract

This paper presents a novel class of systems assisting diagnosis and personalised assessment of diseases in healthcare. The targeted systems are end-to-end deep neural architectures that are designed (trained and tested) and subsequently used as whole systems, accepting raw input data and producing the desired outputs. Such architectures are state-of-the-art in image analysis and computer vision, speech recognition and language processing. Their application in healthcare for prediction and diagnosis purposes can produce high accuracy results and can be combined with medical knowledge to improve effectiveness, adaptation and transparency of decision making. The paper focuses on neurodegenerative diseases, particularly Parkinson's, as the development model, by creating a new database and using it for training, evaluating and validating the proposed systems. Experimental results are presented which illustrate the ability of the systems to detect and predict Parkinson's based on medical imaging information.

**Key Words:** Deep Learning, Convolutional Recurrent Neural Networks, Prediction, Adaptation, Clustering, Parkinson's, Healthcare

## 1. Introduction

Current biomedical signal analysis, including medical imaging, is based on signal processing for feature extraction, segmentation, quantitative and qualitative analysis. Recent advances in Machine Learning and Deep Neural Networks (DNNs) have boosted state-of-the-art performance in all related signal processing tasks. DNNs are the state-of-the-art in machine learning and big data analytics, being used in a large number of applications, ranging from defence and surveillance to human computer interaction and question answering systems [12, 21-22]. DNNs can also be applied as end-to-end-architectures which are composed of different network types and are trained to analyse signals, images, text and other inputs [19, 12]. However, they lack on-line adaptation capability and transparency in decision making. This makes their use difficult in fields such as healthcare, where personalisation and trust are key issues.

The current paper aims at advancing the state-of-the-art, by developing and using DNNs able to perform effective analysis of complex data for healthcare, with focus on neurodegenerative diseases, in particular Parkinson's [3,11]. For Parkinson's disease (PD), we have the required medical support and expertise and a new public dataset, which enables us to design an end-to-end neural architecture and platform that can be adaptable to patient-specific data. We describe a novel DNN system evaluated on a rich public Parkinson's dataset, which can serve as a model for many other related fields.

Whilst Parkinson’s will provide the test-bed for the proposed end-to-end deep neural system, this system will provide an extensible handle for other neurodegenerative diseases. This aligns directly with the Pathway Analysis across Neurodegenerative Diseases described in [16], as ‘there is clinical, genetic and biochemical evidence that similar molecular pathways are met in different neurodegenerative diseases: Alzheimer’s and dementias, Parkinson’s and related disorders, Huntington’s, motor neuron, prion, spinocerebellar ataxia and spinal muscular atrophy’.

The target of this paper is to design and implement end-to-end deep neural architectures that can assist doctors and clinicians in providing improved and more accurate predictions and assessments, while overcoming existing limitations. Focusing on a specific healthcare problem, we design DNN systems integrating imaging, demographic/epidemiological and clinical data, to support doctors in patient-specific prediction and assessment. To achieve this goal, we present a novel approach, developing a combined supervised and unsupervised learning methodology. First, data-driven supervised training of deep neural networks is performed and, then, clustering of the derived network structures is applied to improve the derived results and allow adaptation and handling of new subject cases.

Section 2 presents the new Parkinson’s database, that we have been developing, providing the necessary datasets for training and testing the developed deep neural network systems. Section 3 describes the design of DNN architectures for prediction and diagnosis in healthcare applications. The proposed deep neural systems are based on deep Convolutional (CNN) and Recurrent Neural Networks (RNN), which prove to be able to process all types of available data. A novel methodology for network adaptation when facing new subjects, for personalised assessment, as well as for providing transparency to the network’s performance, is presented in Section 4. An experimental study, illustrating the performance of the generated deep neural architectures, is provided in Section 5. Conclusions and further planned work are given in Section 6 of the paper.

## **2. Generation of the Parkinson’s Database**

We have been creating a novel public dataset composed of 100 patients with Parkinson’s and 40 subjects with Parkinson-related syndromes, including subjects’ MRI, DaT Scans and clinical data. In this paper, we are developing the proposed system based on one third of this dataset, which is the part that has been generated until now. The database is becoming publicly available as *Parkinson Dataset – v1*.

*MRI data:* The rapid evolution of non-invasive medical imaging techniques, over the past decades, has opened new possibilities for the analysis of the brain. The basic imaging technique is Magnetic Resonance Imaging (MRI) which can yield from hundreds to even thousands of images per scan. The assessment of this extremely large set of images per patient can be complicated and time-consuming for doctors. In Parkinson’s Disease, the MRI can show the extent to which the different structures of the brain have been degenerated. Figure 1 shows an example of an MRI. Our main concern regarding Parkinson’s is the Lentiform Nucleus (green line in Figure 2) and the capita of the Caudate Nucleus (red line in Fig. 2). Since

we focus on volume estimation, we process the image sequences in batches, each composed of 3-4 consecutive frames.

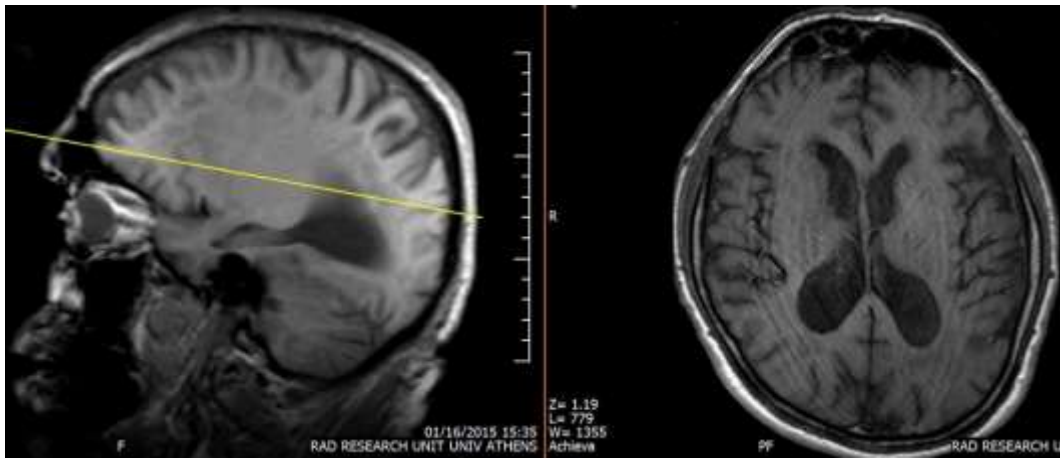


Figure 1. A frame of an axial T1 sequence from a brain MRI (right). Location of the previous slice is placed with regard to a sagittal view of the brain (left).

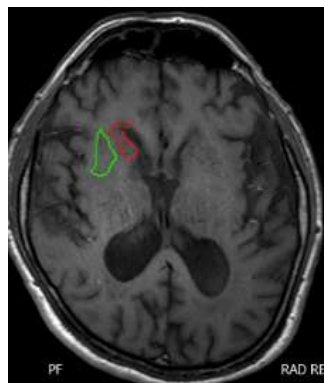


Figure 2. An image from an axial T1 sequence. The Lentiform Nucleus is depicted with a green line, while the capita of the Caudate Nucleus with a red line.

*DaT Scan*: The second brain imaging technique included in the database is Dopamine Transporters (DaT) Scan. This examination is a form of Single-Photon Emission Computer Tomography (SPECT) with Ioflupane Iodide-123 as it's contrast agent. In this examination, we can detect the extent of dopaminergic innervations to the Striatum from the Substantia Nigra. A series of images is produced in this way, as shown in Figure 3.

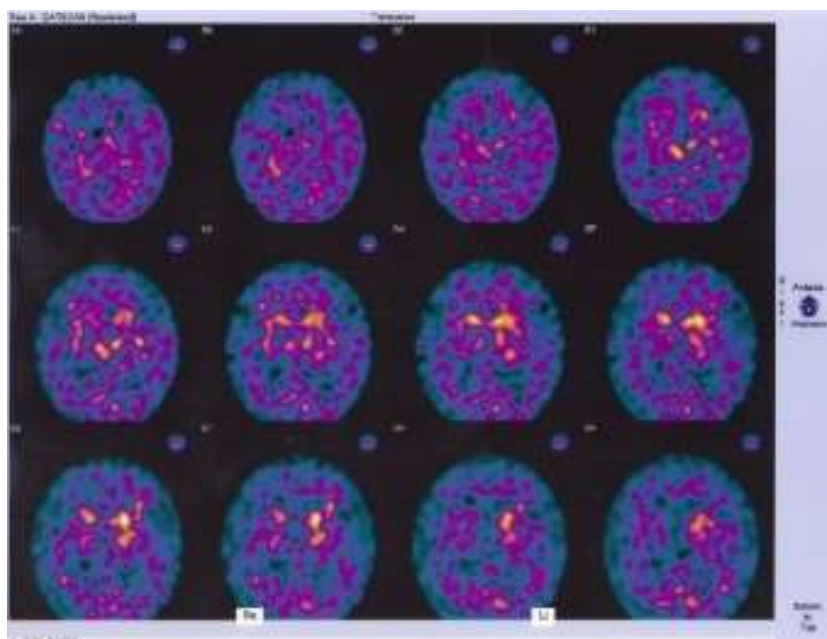


Figure 3. A sequence of frames from a DaT scan.

The doctor selects the most representative ones (the 8<sup>th</sup> in the sequence of Figure 3), and marks the areas corresponding to the head of the caudate nucleus. An automated system then compares these areas with a neutral one (e.g., the cerebellum) and produces the ratios shown at the bottom of Figure 4. Diagnosis is based on comparison of these ratios with normal ones.

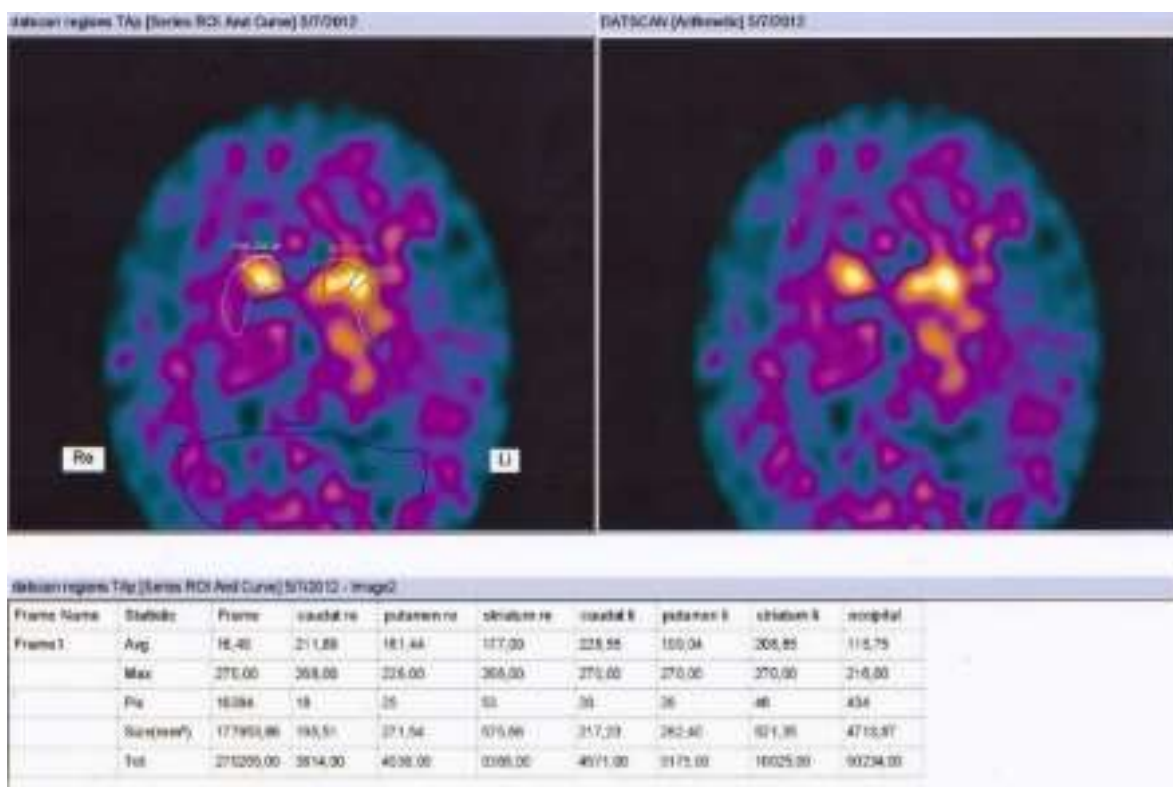


Figure 4. DaT scan with expert selection (left). Same image without the markings (right). Ratios, representing the dopamine deficiency, that are used for the diagnosis (bottom).

*Clinical Data:* These define the patient’s clinical status. We focused on the following scales: UPDRS, the patient’s stage, UDysRS, PDQ-39, FOG, MMSE and two, timed tests [4].

The Unified Parkinson’s Disease Rating Scale (UPDRS) [9] is a metric that examines the patient’s whole clinical performance in 4 parts: motor/non-motor experiences of daily living, motor examination and complications. These contain 13, 13, 18 and 6 elements respectively, with each ranging from 0-4 for a max score of 234.

The patient’s stage [14] represents the evolution of the disease and ranges from asymptomatic (0) to bedridden (5).

The Unified Dyskinesia Rating Scale (UDysRS) [10] was created for evaluating the involuntary movements associated with PD; it has two parts measuring the dyskinesia and dystonia appearing “on” and “off” phases respectively. The first part has 11 while the second 15 elements, all ranging from 0 (asymptomatic) to 4 (severe symptoms), for a total of 150.

The Parkinson’s Disease Questionnaire consists of 39 questions assessing patient’s functionality and quality of life (PDQ-39) [17]. It can be separated into 8 different categories, while each question represents the frequency of a specific incident, ranging from 0 (never occurring) to 5 (always occurring), for a total of 156.

The “Freezing of Gait” (FOG) [8] is one of the most characteristic PD symptoms. The quantification of this symptom is achieved through the homonymous questionnaire which contains 16 elements for a max rating of 24.

The Mini Mental State Examination (MMSE) [24] is an 11-question questionnaire meant to measure the cognitive impairment associated with PD, with a max rating of 30.

Each of MRI and DaT Scan sets includes sequences/multiple scans. For training, we combine annotated data from both types to create thousands of input data, sufficient to train the proposed systems.

### **3. Design of Deep Neural Architectures for Healthcare**

Our main goal is to design deep neural architectures and to evaluate their ability to extract correlations in the available datasets, providing a novel platform for assisting doctors in detecting and assessing disease states. Validation is done using the above-described Parkinson’s dataset. We also target at endowing our system with adaptation capabilities and to test and validate it when handling new patient cases.

The technologies which we use and extend, in order to develop the novel end-to-end deep neural architecture for diagnosis and prediction are:

*Deep Convolutional Neural Networks:* Deep CNNs are architectures that try to exploit the spatial structure of input information [12]. They have been used with great success in various applications, including image analysis, vision, object and emotion recognition. The most successful CNN was used for classifying millions of images in 1000 classes [21].

*Transfer Learning:* Transfer learning [22] is the main approach to avoid learning failure due to overfitting, when training complex CNNs with small amounts of (image) data. In transfer learning, we use networks previously trained with large image datasets (even of generic objects) and fine-tune the whole, or parts of them, using the small training datasets.

*Recurrent Neural Networks:* RNNs are very powerful for processing sequential data [18]. A very successful model, the Long Short-Term Memory (LSTM) [25], uses hidden units with gates that explicitly control data flow in terms of both hidden states and inputs. Bidirectional (B-LSTM) models are obtained by combining forward and backward processing of input data. It is also possible to use Gated Recurrent Units (GRUs) [2, 12] in place of the BLSTM ones. This is explored in the experiments of Section 5.

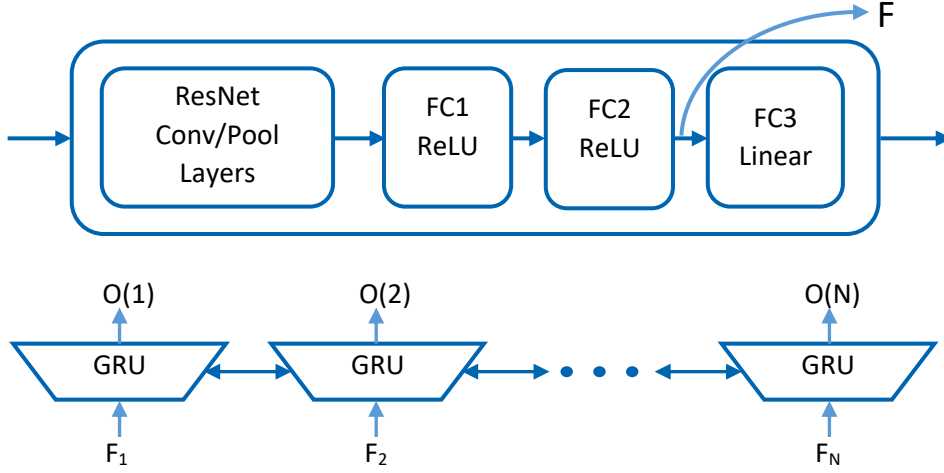


Figure 5. The CNN part of the CNN-RNN architecture feeds the RNN part which yields the final outputs

We propose an end-to-end deep neural architecture including both CNN and RNN components. CNNs derive rich internal representations from input data; B-LSTM/GRU RNNs correlate/analyse time evolution of the inputs, providing the final predictions. CNNs can have a basic structure of the so called VGG-16 network. This CNN model [23] has achieved high accuracy in large image classification problems. It processes images through 13 convolutional layers (core building blocks), 5 pooling layers (performing a form of non-linear down-sampling) and 3 FC (fully connected) layers. In the CNN model, we implemented neurons as Rectified Linear Units (ReLU), a commonly used kind of units employing specific types of non-linear activation functions (rectifiers). Another such structure for consideration is the Deep Residual Net (ResNet) with 152 layers [13]. MRI and DaT Scans are provided at the input of these networks. In fact, it is ResNet that has been mainly used in the experiments reported in this paper. When epidemiological and clinical data values are to be considered, they will be provided directly to the FC1 layer.

Figure 5 shows the CNN-RNN architecture. The CNN part of the neural architecture, using a linear FC3 layer provides continuous clinical data estimation. The CNN feeds the RNN part with the neuron outputs of its second FC layer (F). The RNN accepts  $F_1, F_2, F_3, \dots, F_N$  and delivers predicted values  $O(1), \dots, O(N)$

through time, at its output. A total of 4 images are given to the architecture as a single input. These include 3 grayscale consecutive frames from an axial T1 MRI and a colour DaT scan.

To implement this architecture, we first perform transfer learning of the weights of the convolutional and pooling parts of, e.g., the ResNet network to it. These parts are then fixed during the training phase, where we only train the fully connected layers of the system. The pre-trained convolutional networks have already learnt to generate rich image representations that have proven adequate for image classification and segmentation. These representations are abstract enough to help with specialized tasks, such as the analysis of MRIs and DaT Scans.

This leaves the fully connected part of the network, which is the only part of the network that we actually train in the CNN case. Many variants of this approach have been designed and tested. We selected to freeze the weights of some of the fully connected layers, particularly those belonging to the first FC layer. We have also considered some additional weights of the network as free parameters, by applying fine tuning (a smaller learning rate value) to the weights of (some of) the convolutional layers of the ResNet network, while using a normal learning rate value for the FC part of it.

We use the TensorFlow Platform as the main tool for generating the software implementation of the presented architecture. TensorFlow is a toolkit which got published by Google, under Apache License 2.0. It is mainly implemented using C++, with a significant bit of Python. Its architecture provides the ability to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API.

#### **4. A Novel Method for Deep Neural Network Adaptation and Transparency**

We aim at providing the deep neural architecture with the ability to adapt to new subject cases, assisting doctors with efficient patient-specific analysis and treatment selection, without forgetting its former knowledge. Our methodology is based on a new network retraining approach which extends the work in [5,19]. This approach uses clustering [26] of trained system internal representations, in particular, of the neurons' outputs at the last fully connected CNN layer (denoted, in vector form, as  $F$  in Figure 5), or at the last hidden RNN layer (let us denote them, in vector form, as  $u$ , and consider them feeding the output units  $o$ ). We use the centers of these clusters as knowledge extracted from the data-driven supervised training of the DNN architecture.

Whenever a new subject's data are applied to the input of the DNN end-to-end architecture, the latter computes the respective internal representations and provides a prediction at its output. Our approach is next to compute the distances of these representations from the above described cluster centers and use them to validate, or not, the DNN prediction on these new data. If one of these distances is small, compared to some appropriate threshold, then classification of the new data is made in the same category (patient/non-patient) with that of the specific cluster, generally coinciding with the DNN prediction. If all distances are large, then a drift in the DNN modelling procedure is detected. In the case of drift, we need to train again

the DNN including the new data. However, we do not perform the usual fine-tuning procedure. We choose to retrain the fully connected CNN layers and/or the RNN hidden and output layers, using, on the one hand, the input (image) data corresponding to the cluster centers (Existing Knowledge) and, on the other hand, the new data.

Following this retraining procedure, we avoid the catastrophic forgetting problem in DNN systems, which occurs when we apply repeated fine-tuning to new data cases. This is so, because we keep both the old knowledge (through the cluster centers' information) and the new information provided by specific subject cases. Following retraining, we update the cluster centers as well, after medical validation of the new data, so as to create personalised system knowledge instances.

In particular, the retraining procedure can be implemented as follows:

Let us first consider that, based on the training of the deep neural architecture for Parkinson's, a specific set, say  $S_b$ , including the training input data corresponding to the previously computed cluster centers and the respective annotations (patient/non-patient), has been created. Let  $y(i)$  denote the network output when applied to a new data sample,  $i = 1, 2, \dots$ , not included in the previous network training data set.

Let  $w_b$  include all already computed weights of the fully connected and output layers in a CNN network - and of hidden layers in a CNN-RNN network - before retraining and  $w_a$  the new (updated) weight vector which will be obtained through retraining. In particular, let  $w_b^l$  and  $w_a^l$  respectively denote the weights connecting the outputs of the last hidden layer, say  $u$ , to the network outputs,  $y$ .

A training set  $S_t$  is assumed to include the new input (image) data; this will normally include a rather small number of data.

In the proposed retraining procedure, the new network weights,  $w_a$ , are computed by minimizing the following error criterion:

$$E_a = E_{t,a} + \eta \cdot E_{f,a} \quad (1)$$

where  $E_{t,a}$  denotes the error performed over training set  $S_t$ , i.e., over current input information and  $E_{f,a}$  is the corresponding error performed over training set  $S_b$ , i.e. over previous deep neural network knowledge. Parameter  $\eta$  is a weighting factor accounting for the significance of the current training set compared to the former one. In our approach, we minimize (1) by assuming that a small perturbation of the weights of the fully connected (and/or hidden) layers in the CNN (or CNN-RNN) network is enough to achieve good classification performance in the current conditions. Consequently, we get:

$$w_a = w_b + \Delta w \quad (2)$$

and, similarly,

$$w_a^l = w_b^l + \Delta w^l \quad (3)$$



with  $\Delta w$  and  $\Delta w^l$  being small weight increments. This assumption permits linearization of the nonlinear activation neuron function, using a first-order Taylor series expansion.

It is possible to use the Mean Square Error (MSE) criterion for both quantities in the right-hand side of (1). In this case, we use normal deep learning for CNN and/or RNN networks [12], implemented in the TensorFlow environment. It can be also possible to stress the importance of current data in the minimization of (1). In this case, we replace the first term in the right-hand side of it by the constraint that the actual network outputs  $z_a(i)$ , after retraining, are equal to the desired ones, i.e.,

$$z_a(i) = d(i), \text{ for all data } i \text{ in } S_t \quad (4)$$

Let us denote the difference of the actual network outputs, after and before retraining, in the case of a CNN network, as follows:

$$\Delta z(i) = z_a(i) - z_b(i) \quad (5)$$

Through linearization and using the fact that the outputs  $z$  are weighted averages of the last hidden layer's outputs  $u$ , with the  $w^l$  weights, it can be shown that

$$z_a(i) = z_b(i) + f'_b \cdot w_b^l \cdot \Delta u^l(i) + \Delta w^l \cdot u_b^l(i) \quad (6)$$

where  $f'$  accounts for the derivative of the activation function of the network output neuron(s).

Using Eq. (4) in (6) we get

$$d(i) - z_b(i) = f'_b \cdot w_b^l \cdot \Delta u^l(i) + \Delta w^l \cdot u_b^l(i) \quad (7)$$

All quantities in Eq. (7) are based on former network values, apart from the updates of the weights  $\Delta w^l$  and of the outputs  $\Delta u^l$ . Thus Eq. (7) relates the targeted weights updates in the network output with the outputs of the last hidden layer.

By continuing linearization of the difference of the  $u$  values, towards the previous fully connected layers, we replace the  $\Delta u^l(i)$  term with its equivalent in terms of the weights of the former layers. This continues until we reach the last convolutional layer, which we use with no retraining, and therefore  $\Delta u$  is zero.

In this way, similarly to [5] we compute the weight increments  $\Delta w$  by solving a set of linear equations, over all data in  $S_t$ :

$$c = A \cdot \Delta w \quad (8)$$

with matrix  $A$  being computed in terms of previously trained weights, as was above described, while the elements of vector  $c$  are defined as follows:

$$c(i) = d(i) - z_b(i), \text{ for all data } i \text{ in } S_t \quad (9)$$

and  $z_b(i)$  denotes the outputs of the originally trained network, when this is applied to the data in  $S_t$ .

The size of vector  $c$  is smaller than the number of unknown weights  $\Delta w$ , thus many solutions exist for (8). Uniqueness, however, is imposed by an additional requirement which is to select the solution that causes a minimal degradation of the previous network knowledge. This is of great significance in our approach, since this knowledge (and the respective cluster centers) has been, normally, already validated by medical experts and, therefore, should be changed the least possible.

Thus, the retraining problem results in minimization of (1) subject to constraints (3) and the constraint for small weight increments. A variety of methods can be used for this minimization. One of them is the gradient projection method, which, starting from a feasible point, moves in a direction which decreases the error criterion and satisfies the above constraints. This is used for CNN network retraining in the TensorFlow environment. Extension in the CNN-RNN case is more complex, also taking into account the time evolution and derivatives of the  $u$  values.

In addition to personalized diagnosis and prediction, the proposed approach allows the deep neural architecture to exhibit transparency in its decision making. In particular, for each cluster center, the respective medical input and desired output data are stored in the database, as representative of all data belonging to this cluster. Whenever, upon presentation of new input data to the DNN, the obtained output vector matches that of a specific cluster center, then the respective input image and medical data are presented to the clinician/user to illustrate that this similarity has been taken into account by the network in computing its prediction.

## 5. Experimental Study

The current size of the generated fully annotated database is 45 subjects (about one third of the size to be finally generated), with a ratio of 2:1 between Parkinson’s patients and non-Parkinson’s patients. At this stage, it consists of MRI and DaT scans, annotated as belonging to subjects with Parkinson’s or not.

### 5.1 Dataset generation

We generated a dataset of about 150.000 combinations of color DaT scans with triplets of consecutive MRI gray scale images, for the patient category, and 80.000 such combinations for the non-patient category. Each input (combination) consists of three MRI images and one RGB DaT scan image. To obtain a balanced dataset, we applied various augmentation techniques, such as over-sampling the latter category, or under-sampling the former [1]. The above were then used as training data for designing the end-to-end deep neural architectures.

Moreover, we kept the data of about 15% of the subjects for validation/testing. It should be emphasized that our target has been to test the ability of the networks to learn from a number of patients and generalize their performance to other subjects, who have not been included in the training set. For this reason, the test data consisted of six new subjects, four with Parkinson’s (PD patients) and two without (Non-PD patients,

denoted NPD), to provide about 1.200 test input samples. The networks had two linear outputs, with targeted values (1,0) and (0,1), respectively, for the two categories.

As a reference, 10 consecutive frames from an axial T1 brain MRI are presented in Figure 6 for a patient without Parkinson's, and 10 more in Figure 7 for a patient with the disease.

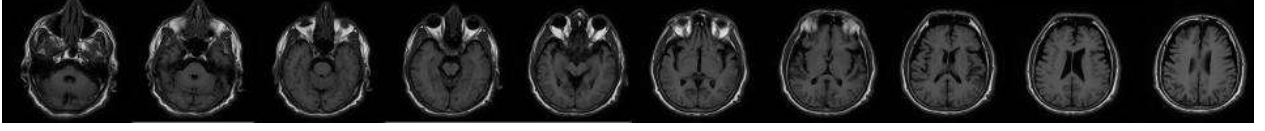


Figure 6. MRI scan of a patient without Parkinson's Disease.  
Axial orientation - T1 sequence.

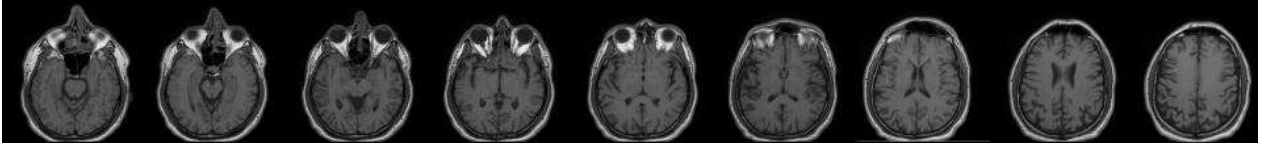


Figure 7. MRI scan of a patient with Parkinson's Disease.  
Axial orientation - T1 sequence.

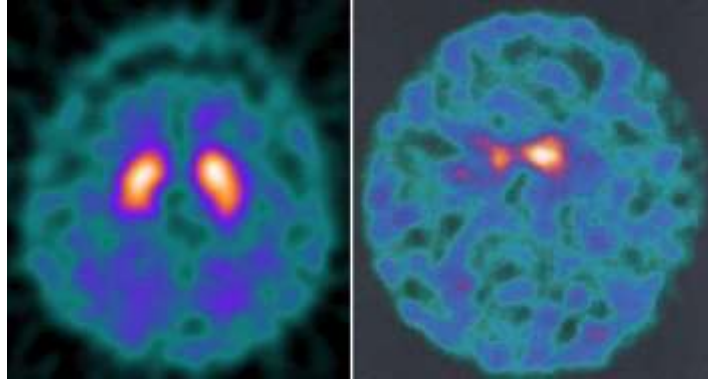


Figure 8. DaT scan from a patient without Parkinson's Disease (left).  
Respective image from a patient with Parkinson's (right)

Figure 8 shows two DaT scans of patients without and with Parkinson's Disease, respectively. The dopamine deficiency can be seen in these images.

## 5.2 Network training

As a first approach, we selected to train the CNN and CNN-RNN deep neural networks from scratch; starting from random initial weights in the convolutional and fully connected (FC) parts of the CNNs, or the convolutional and hidden layers of the CNN-RNNs. As a second approach, we adopted transfer learning,

i.e. transfer of the weights of the convolutional and pooling layers of a pretrained CNN, to the generated networks. Then, the ‘upper’ FC part of the targeted CNN network, as well as the RNN hidden layers of the CNN-RNN, were designed and trained with the above dataset. For the initialization of these weights, we used the ResNet-50 CNN, which has been pre-trained with millions of general type RGB images for this purpose. A separate system was used for each of the image types in our inputs, i.e., one focusing on the MRI triplets and another focusing on the DaT scan. We concatenated the outputs of these two ResNet substructures at the input of the first FC layer of the CNN network. It is at this layer, that epidemiological data will be concatenated as well, when the whole database will have been generated.

Based on this procedure, we separately trained both a deep CNN network and a deep CNN-RNN network for Parkinson’s disease diagnosis.

### 5.3 Experimental evaluation

Table 1 summarizes the results obtained through different configurations of the CNN network, i.e. ones with different numbers of hidden layers and hidden units per layer. An accuracy of 96% on training and 94% on testing datasets was obtained in this experiment, which is very satisfactory.

Table 1. Performance of the End-to-end CNN Architecture for Parkinson’s

CNN architectures: 2 output units (PD/NPD)	Number of Fully Connected (FC) Layers	Number of Units in each FC Layer	Accuracy
1	1	1000	0.57
2	1	2622	0.60
3	2	2622-500	0.90
4	2	2622-1000	0.91
5	2	2622-1500	0.94
6	2	2622-2000	0.93

Table 2 summarizes the accuracy obtained by the CNN-RNN (with GRU neuron model) architecture, for different respective structures. The addition of the RNN part allows the deep neural architecture to better follow time varying correlations in the MRI sequence of triplets of frames, thus increasing the accuracy of Parkinson’s prediction to 98% on the testing data set.

Table 2. Performance of the End-to-end CNN-RNN Architecture for Parkinson's

CNN-RNN architectures: 1 Fully Connected Layer 2 Hidden Layers (128 Units each) 2 (linear) output units	Number of Units in the FC Layer	Accuracy
1	500	0.91
2	1000	0.96
3	1500	0.98
4	2000	0.97

There are some additional metrics obtained in terms of the above results. In the best reported case (line 3 of Table 2), the MSE value was very low, equal to 0.02. Considering the binary problem examined in this paper (PD/NPD), precision attained 1.00 and recall 0.96 (F1 value 0.98).

Figures 9 and 10 show the accuracy obtained by the end-to-end deep CNN and CNN-RNN architectures, respectively, on the validation/test data set, during training. It can be shown that the best accuracy of the CNN architecture is obtained early in the learning phase, afterwards reaching overfitting conditions. It can also be observed that the Deep CNN-RNN architecture takes longer to derive the best performance than the CNN one.

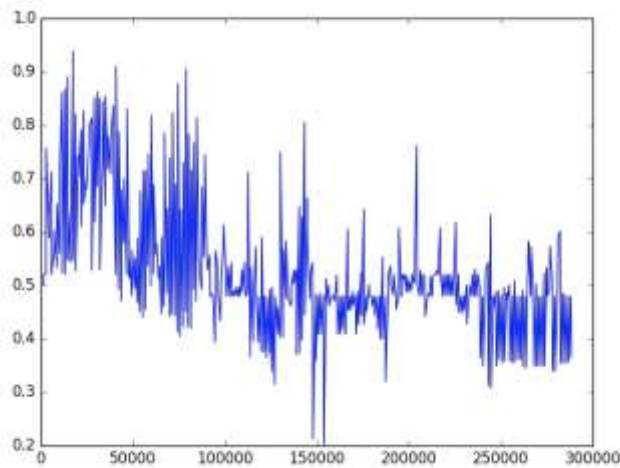


Figure 9. CNN Performance on validation data, during training epochs

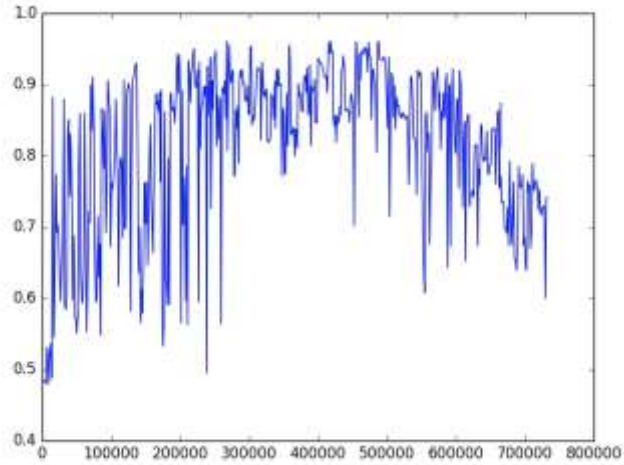


Figure 10. CNN-RNN Performance on validation data, during training epochs

It should be mentioned that the best performance of the CNN-RNN architecture was 99,97% on the training data and 98% on the test data; the latter consisted of about 1200 input data from six subjects, none of their data having been included in the training data set. About 600 data concerned each one of the PD and NPD categories. The performance on test data was 96% for PD and perfect, i.e., 100%, for NPD patients. In particular, Table 3 shows the percentage of correct classifications for each test subject's data (combinations of MRIs and DaT scans).

Table 3. Testing performance of trained CNN-RNN Architecture on each subject

Subject number in the database	Category	Correct Classifications (normalized [0,1])
26	PD	0,90
4	PD	1,00
6	PD	0,985
9	PD	0,956
17	NPD	1,00
21	NPD	1,00

It can be seen that this is an excellent result, which shows the potential of the deep CNN-RNN architecture to provide very accurate predictions of Parkinson's disease.

We then applied the proposed clustering procedure on the representations (vector of neuron outputs) generated at the last hidden layer of the trained CNN and CNN-RNN. This produced various cluster sizes on the training data set. We chose to discuss here the clustering of the representations obtained when the deep neural architectures are applied to the above test data, since we can visually illustrate the obtained results, also relating them to the performances presented in Table 3.

#### 5.4 Clustering visualization

In order to visually illustrate the distribution of data in categories, Principal Component Analysis (PCA) was performed on the representations obtained through processing of the test data. Focus was put on the derived two main principal components, as shown in Figures 11a and 12a, for the CNN and CNN-RNN architectures, respectively.

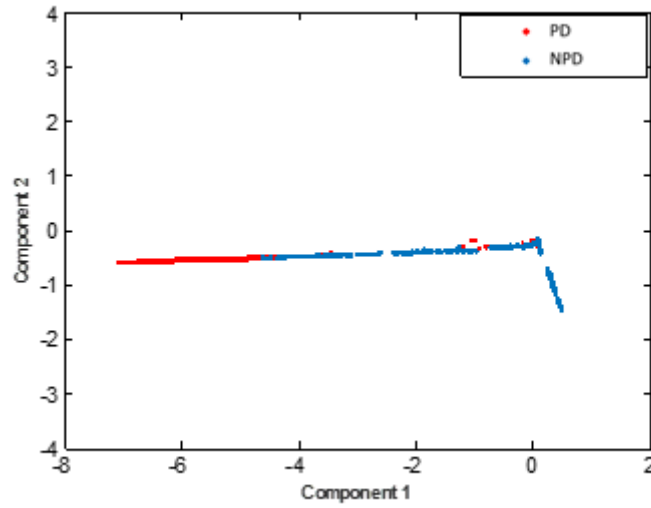


Figure 11a. The two main principal components of the CNN representation.

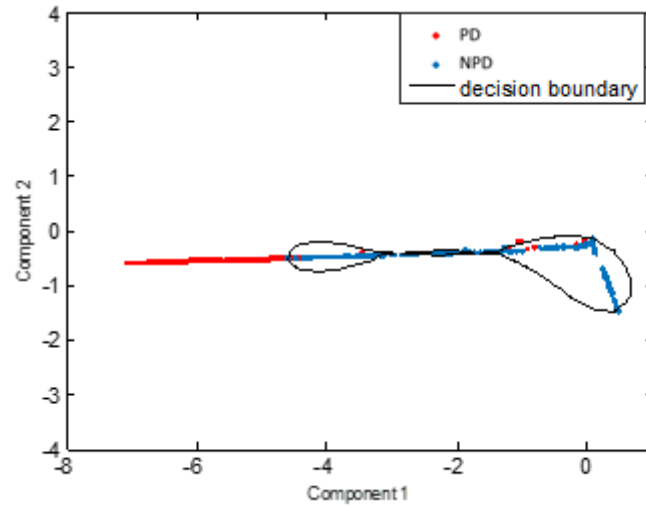


Figure 11b. Visualization of (three) cluster boundaries for the NPD category provided by an OCSVM approach.

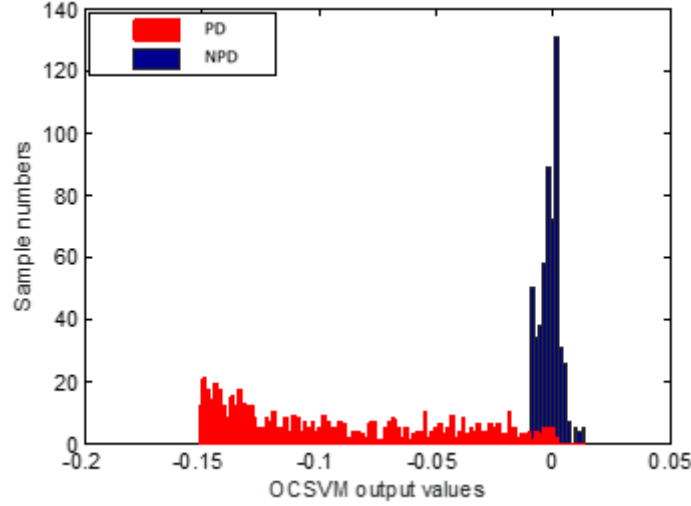


Figure 11c. Histogram of the derived OCSVM outputs.

Figure 11a shows the distribution of the representations obtained for PD and NPD subjects, as derived from the CNN architecture. It should be mentioned that the last CNN fully connected layer consisted of 1500 neurons. However, due to the ReLU activation function, only about 30 neurons yielded non-zero values in this representation. Figure 11b verifies the ability of a one-class support vector machine (OCSVM) [26], to determine clusters corresponding to the NPD class. Three classes are shown in Figure 11b. We were able to get six classes per category, to reach the performance of the DNN in this case.

It is interesting to mention the variability of the PD cases compared to the NPD ones. This is in accordance with the lower accuracy obtained by the DNN architecture in the PD class, when compared to the NPD case. Figure 11c shows a histogram of the OCSVM values also illustrating this observation.

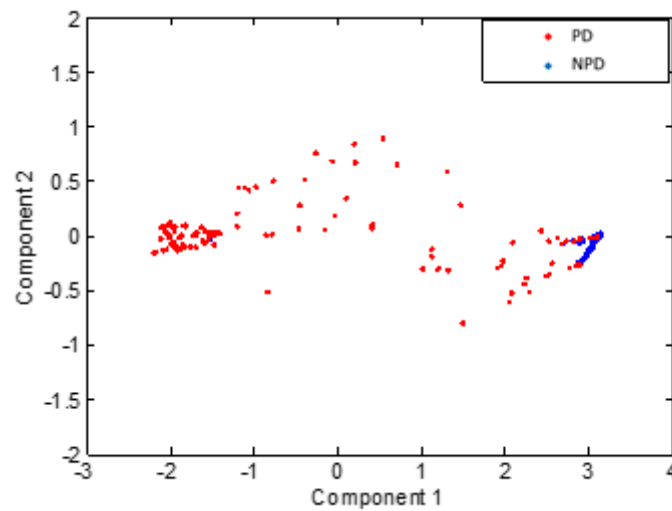


Figure 12a. The two main principal components of the CNN-RNN representation.



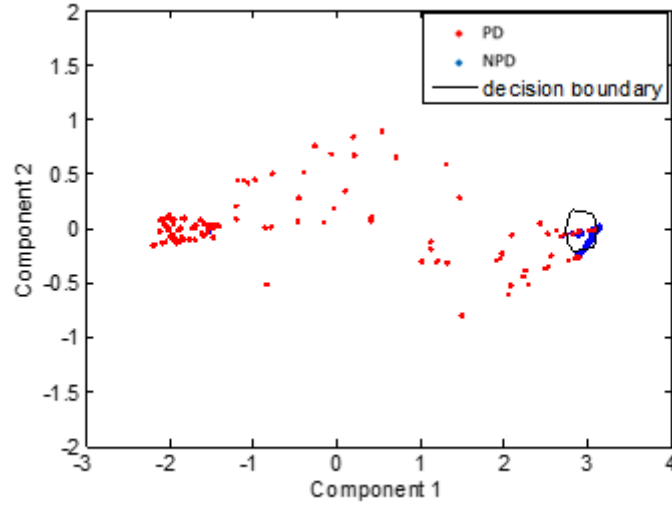


Figure 12b. Visualization of (one) cluster boundary for the NPD category provided by an OCSVM approach.

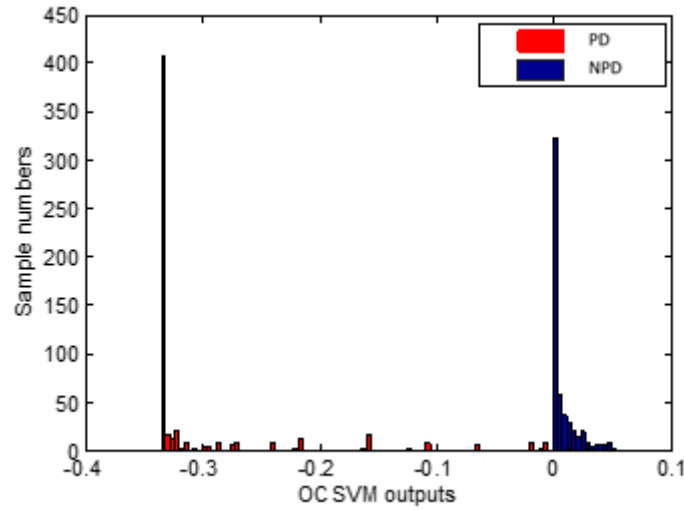


Figure 12c. Histogram of the derived OCSVM outputs.

The respective results obtained for the representations provided by the CNN-RNN architecture are shown in Figures 12a-c. It should be mentioned that, in this case, the obtained representations consisted of 128 neuron output values, computed through the *tanh* activation function. However, only about 20 of the neurons provided significant non-zero values; the rest yielded very small, practically negligible, values.

By comparing these results with the respective ones in Figures 11a-c, it is concluded that the CNN-RNN architecture - which has achieved a better performance than CNN- has been able to produce much more compact representations for each category, with well separated clusters.

An indication of the purity (through precision) of the clusters in the augmented training set can be viewed in Table 4.

Table 4. Cluster precision on the training set

Cluster	1	2	3	4	5
Category					
PD	0	5	18277	1516	18163
NPD	2822	25393	0	0	0

We computed the cluster centers, as the mean values of all 128-dimensional vector representations included in each cluster. Their projection in 3-D is shown in Figure 13, showing the significant distances between them. Moreover, Table 5 shows the corresponding maximum mean square distance of the representations in each cluster from the corresponding cluster center.

Table 5. Maximum intra-cluster distance (from the center)

Cluster	1	2	3	4	5
Distance (MSE)	0,01	0,02	1,565	0,158	0,14

Figures 15a-e illustrate the input images corresponding to the 5 cluster centers that were derived from the CNN-RNN architecture. The clusters have been sorted by the level of degeneration of the basal ganglia (Lentiform Nucleus, Caudate Nucleus). These five clusters roughly represent the 3 stages of DaT loss in PD, as confirmed by medical experts.

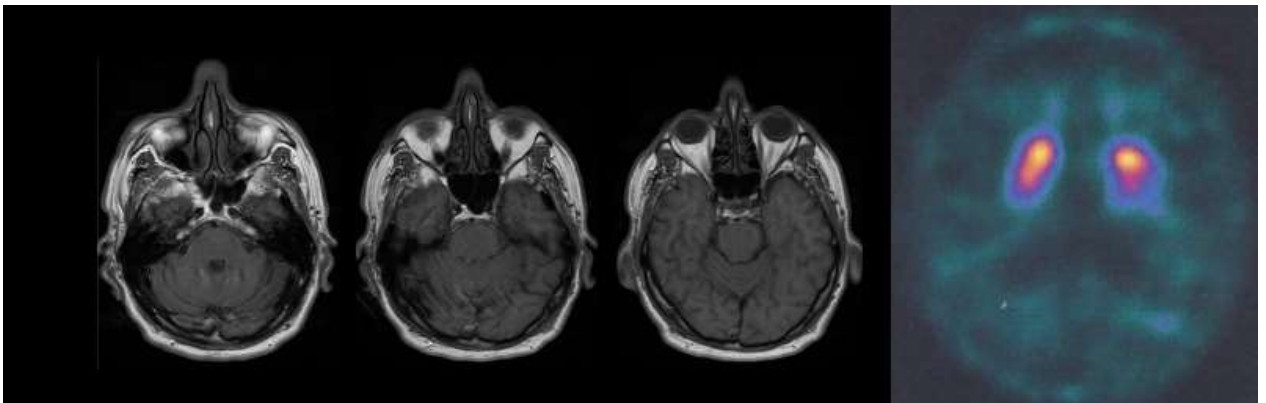


Figure 15a. The first cluster center corresponds to a typical frame from a DaT scan of an individual not suffering from PD.

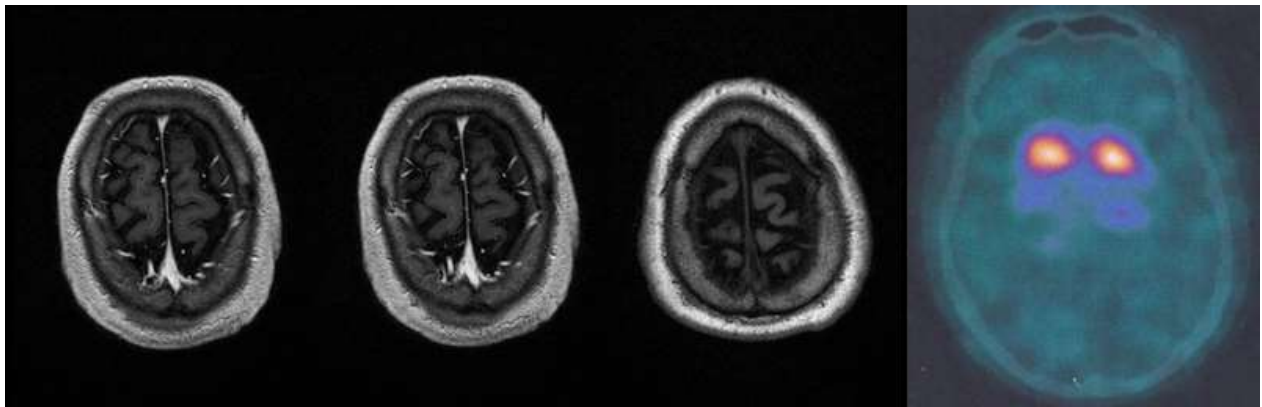


Figure 15b. The second cluster center represents an interesting case of an image that seems to be pathological but belongs to a healthy individual. Though the Lenticular Nucleus appears to be completely gone, there is no diffusion of the contrast agent in the brain. The latter could be viewed as an indication that the main structures are, in fact, intact.

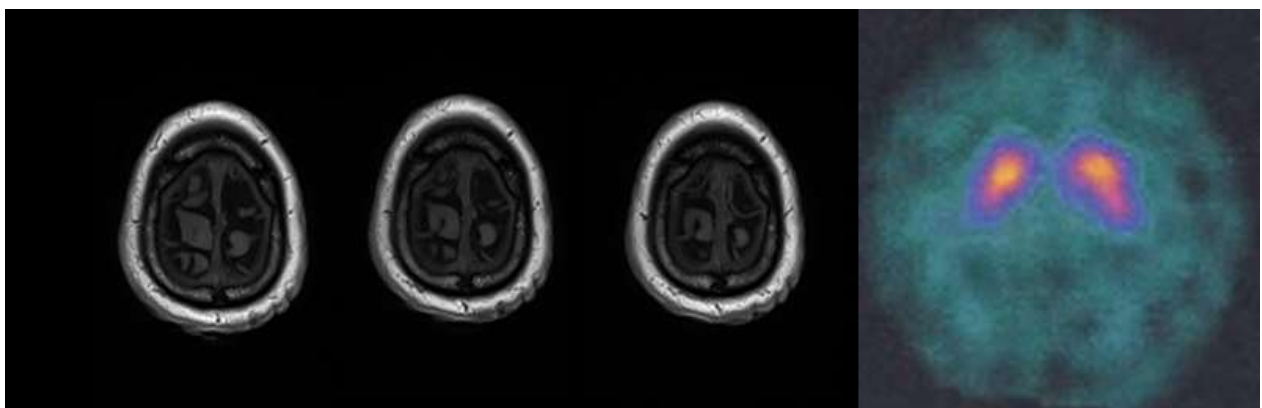


Figure 15c. The third cluster represents the early stages (1-2) of the degeneration associated with PD, as both Lenticular Nuclei appear to be diminishing.

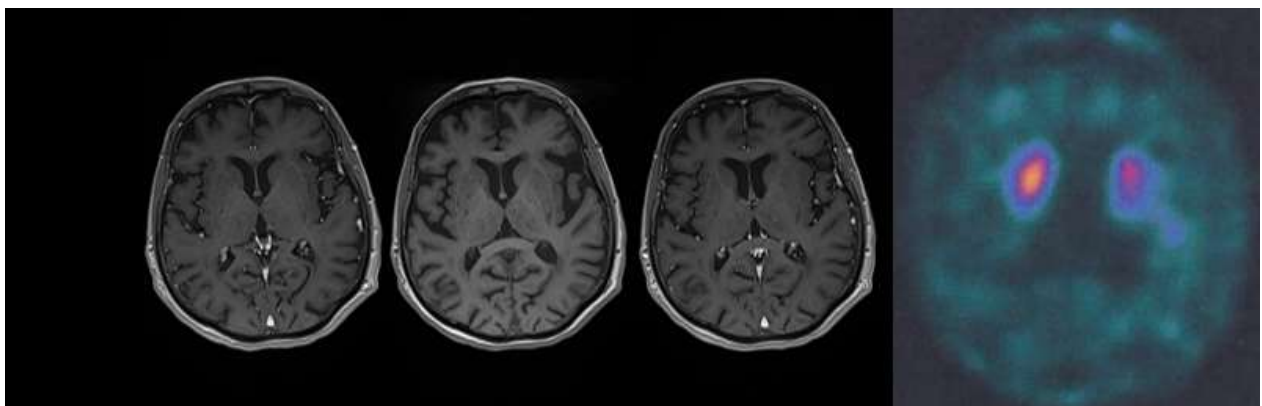


Figure 15d. The fourth cluster is a typical stage 2 DaT loss. Both Lenticular Nuclei are completely gone; the only signal is from the caudate, which appear as two almost symmetrical circular areas.

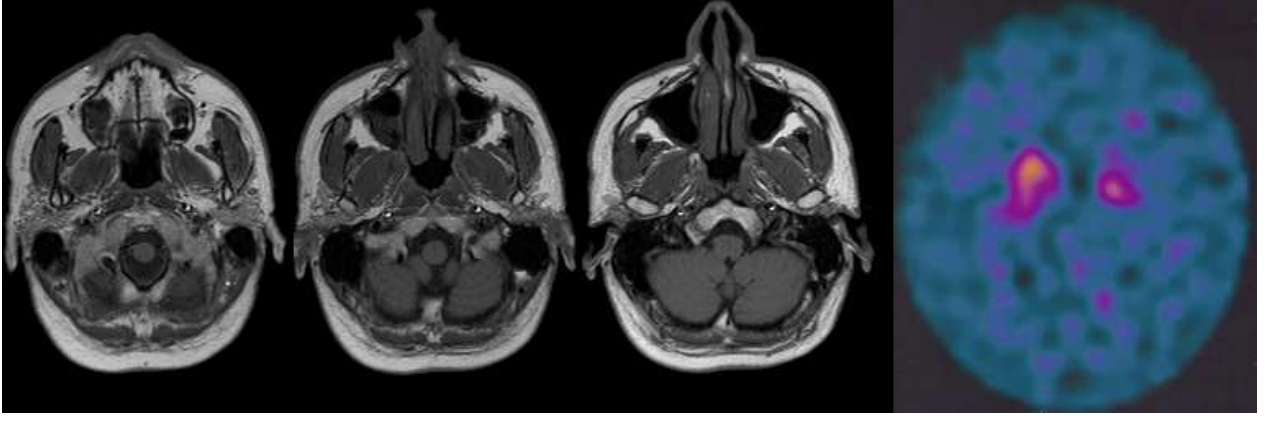


Figure 15e. The fifth cluster is the most advanced stage of DaT loss – stage 3. Here the basal ganglia appear further degenerated, while there is significant activity in the rest of the brain. This is an indication that these structures have lost their ability to contain the contrast agent and it has diffused throughout the brain.

It is with this end-to-end deep neural architecture and the derived representations, that we further investigate the obtained results in Table 3. In particular, we consider these cases as new subjects' data, presented to the trained CNN-RNN for prediction and diagnosis. Since these cases span different possible scenarios, we will evaluate them in two different steps.

Let us first consider, the 4, 17 and 21 subjects of Table 3 (one from the PD category and two of the NPD category), all data of whom were correctly predicted (100% accuracy) by the CNN-RNN architecture. The internal representations (128-dimensional vectors) generated at the output of the second hidden layer of the RNN were also correctly classified, based on their distances from the centers of the clusters derived from the trained CNN-RNN respective internal representations. All classifications provided by the trained DNN architecture for the data of these three subjects have been, therefore, accepted by our derived end-to-end contextualization approach and formed the finally obtained predictions.

Since the training database has now been increased with three new subject datasets, we perform an updating of the centers of the clusters to which the new data have been included. Let us assume that a single vector  $m[j]$ ,  $j=1, 128$ , is used to update the center  $c_i$  of the  $i$ -th cluster composed of  $N_i$  members. Then, the new class center  $c_{i, new}$  will be slightly modified, as follows:

$$c_{i, new}[j] = N_i \cdot c_{i, old}[j] / (N+1) \quad (10)$$

Consequently, an updated, very slightly different, system memory is produced, incorporating the new knowledge about the new subjects' data.

Let us now focus on the three remaining cases of Table 3, all referring to PD patients. For complete DNN updating, we re-trained the CNN-RNN network, through the procedure described in Eqs. (1)-(6) of the previous section.

10 input combinations, out of 120, 3 out of 204 and 8 out of 184 input combinations, respectively, have been erroneously classified, as NPD cases, both by the CNN-RNN architecture and the cluster-based

representation. We should mention that, since these cases constitute only a 9%, 1,5% and 4,5% of the data obtained by each of these patients, respectively, it will not be difficult for the clinician to examine them in the context generated by all the other images, which have been correctly classified, and provide his/her own diagnosis.

Following this validation, two new clusters have been added to the existing ones, so as to model these new cases. Finally, we used the adaptation methodology described in Section 4 to successfully retrain the DNN architecture so as to accurately classify the new data as well.

In all the above experiments, for DNN training, we used the Adam optimizer algorithm, in mini batches, considering the Mean Squared Error (MSE) as cost function.

### **5.5 Hyper-parameter value selection**

For the CNN architecture, the hyper-parameter values were selected as follows: a batch size of 30 (15 examples from each category), a constant learning rate of 0.001; 2622 and 1500 hidden units respectively in each fully connected layer and dropout after each fully connected layer with a value of 0.5. We also used biases in the fully connected layers.

For the CNN-RNN architectures the hyper-parameters were selected to match the previous ones, apart from the batch size which was 40 (20 examples from each category) and the number of hidden units in the GRU layers, both of which were 128.

The weights of the fully connected layers were initialized from a Truncated Normal distribution with a zero mean and a variance equal to 0.1 and the biases were initialized to 1.

Training was performed on a single GeForce GTX TITAN X GPU and the training time was about 2-3 days.

## **6. Conclusions and Further Work**

We have designed novel end-to-end deep neural architectures, composed of CNN and RNN components, appropriately trained with medical imaging data, and have obtained very good performances in diagnosis and prediction of Parkinson's disease. We have been developing and publicizing a new database, which we have used for training and evaluating the performance of the new deep neural architectures.

Moreover, we have proposed a novel unsupervised approach, based on clustering of the trained DNN internal representations, which provides the deep neural architecture with the ability to adapt to new data cases, without suffering the catastrophic forgetting problem, usually met in DNN fine-tuning adaptation methodologies. This procedure also provides a type of transparency in the decision making process implemented by the deep neural architecture.

In our current research, with the aid of medical experts, we correlate the generated clusters with the medical and clinical data and try to create descriptions relating the DNN decisions with the developed cluster characteristics. This will be the basis for providing explanations of the network's performance, thus, rendering its use transparent and trustful.

A lot of research has been made on neuro-symbolic learning and reasoning, i.e., merging neural networks with knowledge representation, also involving deep neural networks [20, 7] and on extracting rules from trained networks [15]. We will also investigate the use of these methods to provide formal representations of the generated Parkinson's knowledge and/or extract additional rules that may further justify the predictions and assessments of the designed deep neural architectures.

Our future research aims at extending the developments obtained for the Parkinson's case to other degenerative diseases, which are based on similar input medical imaging information. We first target dementias and Alzheimer's, using a recently presented database in [6]. Following the approach proposed in the paper, we will use transfer learning to retrain the DNNs designed for Parkinson's on datasets describing other diseases.

## References:

- [1] Chawla, Nitesh V., Nathalie Japkowicz, and Aleksander Kotcz. Editorial (2004): special issue on learning from imbalanced data sets. *ACM Sigkdd Explorations Newsletter* 6.1: 1-6.
- [2] Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.
- [3] DeMaagd, George, and Ashok Philip (2015): Parkinson's Disease and Its Management: Part 1: Disease Entity, Risk Factors, Pathophysiology, Clinical Presentation, and Diagnosis. *Pharmacy and Therapeutics* 40.8 504.
- [4] Defer, G. L., Widner, H., Marié, R. M., Rémy, P., & Levivier, M. (1999). Core assessment program for surgical interventional therapies in Parkinson's disease (CAPSIT-PD). *Movement Disorders*, 14(4), 572-584.
- [5] Doulamis A, Doulamis N, Kollias S. (2000). On-Line Retractable NNs: Improving the Performance of NNs in Image Analysis Problems. *IEEE Transactions on NNs*, vol. 11, no 1, pp. 137-156, 2000.
- [6] Gao X., Hui R., Tian Z. (2017). Classification of CT Brain Images based on DNNs. *Computer Methods and Programs in Biomedicine*, vol. 138, pp. 49-56.
- [7] Garcez A. d' Acilla et al. (2015). *Neural-Symbolic Learning and Reasoning: Contributions and Challenges*. AAAI Spring Symposium, Stanford University, CA.
- [8] Giladi, N., Shabtai, H., Simon, E. S., Biran, S., Tal, J., & Korczyn, A. D. (2000). Construction of freezing of gait questionnaire for patients with Parkinsonism. *Parkinsonism & related disorders*, 6(3), 165-170.
- [9] Goetz, C. G., Tilley, B. C., Shaftman, S. R., Stebbins, G. T., Fahn, S., Martinez-Martin, P., ... & Dubois, B. (2008). Movement Disorder Society-sponsored revision of the Unified Parkinson's Disease Rating Scale (MDS-UPDRS): Scale presentation and clinimetric testing results. *Movement disorders*, 23(15), 2129-2170.
- [10] Goetz, C. G., Nutt, J. G., & Stebbins, G. T. (2008). The unified dyskinesia rating scale: presentation and clinimetric profile. *Movement Disorders*, 23(16), 2398-2403.
- [11] Goldman SM, Tanner C. Etiology of Parkinson's disease (1998). In: Jankovic J, Tolosa E (eds). *Parkinson's disease and movement disorders*. 3rd ed. Baltimore: Williams and Wilkins: 1998:133-158.

- [12] Goodfellow, I. (2015). Deep learning. *Nature*, 521, 436-444.
- [13] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770-778).
- [14] Hoehn, M. M., & Yahr, M. D. (1967). Parkinsonism onset, progression, and mortality. *Neurology*, 17(5), 427-427.
- [15] Hu, Z. Harnessing Deep NNs with Logic Rules. arXiv:1603.06318v4, 2016.
- [16] JPND EU Joint Programme (2017). Neurodegenerative Disease Research. Pathways, <http://jpnd.eu>.
- [17] Jenkinson C., Fitzpatrick R., Peto V., Greenhall R., Hyman. N. (1997) The Parkinson's Disease Questionnaire (PDQ-39): development and validation of a Parkinson's disease summary index score. *Age and ageing*, vol. 26 no 5, pp. 353-357, 1997.
- [18] S. E. Kahou (2015). Recurrent NNs for emotion recognition in video. *Proc. ACM ICMI*, 2015.
- [19] Kollias D , Tagaris T , Stafylopatis A (2017). On Line Emotion Detection Using Retractable Deep NNs. *IEEE Symposium Series Computational Intelligence 2016*, IEEE Xplore 13-2-2017.
- [20] Kollias D, Marandianos G, Stafylopatis A (2015). Interweaving Deep Learning and Semantic Techniques for HCI. *10th Intern. Workshop on Semantics and Adaptation*. Trento, Italy, 2015.
- [21] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).
- [22] Ng, H. W., Nguyen, V. D., Vonikakis, V., & Winkler, S. (2015, November). Deep learning for emotion recognition on small datasets using transfer learning. In *Proceedings of the 2015 ACM on international conference on multimodal interaction* (pp. 443-449). ACM.
- [23] K. Simonyan et al. CNNs and Large-scale Image Recognition (IR). arXiv:1409.1556, 2014.
- [24] Tombaugh, T. N., & McIntyre, N. J. (1992). The mini-mental state examination: a comprehensive review. *Journal of the American Geriatrics Society*, 40(9), 922-935.
- [25] Wöllmer, M., Kaiser, M., Eyben, F., Schuller, B., & Rigoll, G. (2013). LSTM-modeling of continuous emotions in an audiovisual affect recognition framework. *Image and Vision Computing*, 31(2), 153-163.
- [26] Yu M. (2013). An on-line one class SVM-based person-specific fall detection system. *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 6, pp. 1002–1014.